A New Bandwidth Adaptive Non-Local Kernel Regression Algorithm For Image/Video Restoration and Its GPU Realization

C. Wang and S. C. Chan

Department of Electronic and Electrical Engineering
The University of Hong Kong, Hong Kong
{cwang, scchan}@eee.hku.hk

***Abstract*—This paper presents a new bandwidth adaptive non-local kernel regression (BA-NLKR) algorithm for image and video restoration. NLKR is a recent approach for improving the performance of conventional steering kernel regression (SKR) and local polynomial regression (LPR) in image/video processing. Its bandwidth, which controls the amount of smoothing, however is chosen empirically. The proposed algorithm incorporates the intersecting confidence intervals (ICI) bandwidth selection method into the framework of NLKR to facilitate automatic bandwidth selection so as to achieve better performance. A parallel implementation of the proposed algorithm is also introduced to reduce significantly its computation time. The effectiveness of the proposed algorithm is illustrated by experimental results on both single image and videos super resolution and denoising.**

I. INTRODUCTION

Image and video restoration is an important problem in image and video signal processing. For instance, with better display devices with higher resolutions (HD, etc.), there is a need to convert lower resolution videos to higher spatial resolution and frame rates. Moreover, the needs for supporting high quality close-up or scaling of important image and video objects call for better super-resolution techniques. Due to the various degradations of the original image or video, such operations may need to take these degradations like sensor noise and deblurring into account.

Local kernel regression using say polynomial modeling has recently emerged as a flexible and effective framework for restoration tasks such as image denoising, super-resolution and inpainting [1-9]. For instance, a Steering Kernel Regression (SKR) algorithm [1, 2] for image super-resolution (SR) has been proposed. Local polynomial regression (LPR) or modeling has also been applied to image denoising and inpainting in [3-8]. Both LPR and SKR explore the local smoothness of the image/videos and represent the image/video locally as a polynomial. By estimating the coefficients of such local polynomials, one may interpolate image locally using these local representation and remove high frequency noise. To further exploit the structural information in natural images which may result from repeated structures in the same image, Non-Local Kernel Regression (NLKR) algorithm [9] has been proposed recently. It is also based on the local polynomial representation. However, it utilizes non-local self-similarity in the image to collect similar but distant observations to estimate the local polynomial models. Therefore, it can considerably improve the performance of SKR in presence of noise. Due to the nature of block matching used to find similar blocks, motion estimation is not required for NLKR and it can work for image and video data. Moreover, NLKR also offers promising performance for single image SR due to the introduction of non-local similarity.

An important problem in SKR, LPR and related techniques is the selection of an appropriate bandwidth for smoothing. Too large a bandwidth will result in over smoothing while too small a bandwidth may not provide sufficient attenuation of image noise. Both the SKR and NLKR rely on user supplied empirical bandwidth and hence it is highly desirable to develop automatic bandwidth selection for these methods. In this paper, we propose a Bandwidth Adaptive Non-Local Kernel Regression (BA-NLKR) algorithm for image and video restoration using the intersecting confidence intervals (ICI) bandwidth selection method. To further reduce the computational time of both the original NLKR and the ICI bandwidth selection rule, the whole algorithm has been parallelized on Graphic Processing Units (GPUs). Simulation results show that the proposed BA-NLKR has a higher PSNR than the conventional NLKR algorithm and its GPU implementation also offers significant speedup over conventional CPU implementation.

The rest of the paper is organized as follows. We first briefly review the local and nonlocal kernel regressions in Section II. We then extend the ICI rule for automatic bandwidth selection for NLKR. Its GPU implementation will be briefly outlined in Section IV. Practical applications and experimental results on super-resolution (SR) and denoising will be given in Section V. Finally, we conclude the paper in Section VI.

II. LOCAL AND NON-LOCAL KERNEL REGRESSION

A. Locally Adaptive Kernel Regression

In LPR, the image or video at location $\boldsymbol{x}$ is modeled locally as a polynomial. Let y_j , $j=1,...,p,$ be the observation at location $\boldsymbol{x}_j$ within a small neighborhood $N(\boldsymbol{x}_i)$ of the

This work was supported in part by the General Research Fund (GRF) of Hong Kong Research Grant Council (RGC).

location $\boldsymbol{x}_i$ of interest, i.e. $\mathbf{x}_j \in N(\mathbf{x}_i)$. Ideally, the neighborhood $N(\boldsymbol{x}_i)$ should be chosen so that the image/video can be approximated well by the given polynomial. It was shown in [1] that the local polynomial coefficients can be obtained by weighted least squares (WLS) fit of the given polynomial to the observations, which gives:

$$\hat{\boldsymbol{b}}_i = \arg\min_{\boldsymbol{b}_i} \| \boldsymbol{W}_{K_i} (\boldsymbol{y}_i - \boldsymbol{A}_i \boldsymbol{b}_i) \|_2^2 , \tag{1}$$

where $\boldsymbol{b}_i$ is the vector of polynomial coefficients to be determined, $\boldsymbol{y}_i = [y_1, ..., y_p]^T$ is the vector of observations at $\boldsymbol{x}_i$,

$$\mathbf{A}_i = \begin{bmatrix} 1 & (\mathbf{x}_1 - \mathbf{x}_i)^{\mathrm{T}} & vech^{\mathrm{T}}\{(\mathbf{x}_1 - \mathbf{x}_i)(\mathbf{x}_1 - \mathbf{x}_i)^{\mathrm{T}}\} & \cdots \\ 1 & (\mathbf{x}_2 - \mathbf{x}_i)^{\mathrm{T}} & vech^{\mathrm{T}}\{(\mathbf{x}_2 - \mathbf{x}_i)(\mathbf{x}_2 - \mathbf{x}_i)^{\mathrm{T}}\} & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ 1 & (\mathbf{x}_p - \mathbf{x}_i)^{\mathrm{T}} & vech^{\mathrm{T}}\{(\mathbf{x}_p - \mathbf{x}_i)(\mathbf{x}_p - \mathbf{x}_i)^{\mathrm{T}}\} & \cdots \end{bmatrix}, \tag{2}$$

$\boldsymbol{W}_{K_i} = diag[K_h(\boldsymbol{x}_1 - \boldsymbol{x}_i), K_h(\boldsymbol{x}_2 - \boldsymbol{x}_i), \cdots, K_h(\boldsymbol{x}_p - \boldsymbol{x}_i)]$ is the weight matrix defined by a kernel function $K_h(u) = \frac{1}{h} K(\frac{u}{h})$ which puts more emphasis to observations near $\boldsymbol{x}_i$, $K(u)$ is the kernel function, h is the bandwidth of the kernel which control the size of the neighborhood, and $vech\{\cdot\}$ is the half-vectorization operator. The first element of $\hat{\boldsymbol{b}}_i$ is the smoothed pixel value of at $\boldsymbol{x}_i$ which is given by

$$\hat{z}(\boldsymbol{x}_i) = \boldsymbol{e}_1^{\mathrm{T}} (\boldsymbol{A}_i^T \boldsymbol{W}_{K_i} \boldsymbol{A}_i)^{-1} \boldsymbol{A}_i^T \boldsymbol{W}_{K_i} \boldsymbol{y}_i , \tag{3}$$

where $\mathbf{e}_1 = [1, 0, \cdots, 0]^{\mathrm{T}}$. Note that $\boldsymbol{x}_i$ can be any location in the image and hence the image can be interpolated to a higher resolution. Since the local structure is usually anisotropic, the following locally adaptive kernel has been proposed in [1, 2], which can better adapt to the local gradient of the image:

$$K_h(\boldsymbol{x}_n - \boldsymbol{x}_i) = \frac{\sqrt{\det(\boldsymbol{C}_n)}}{2\pi h^2} \exp(-\frac{1}{2h^2} \| \boldsymbol{C}_n^{1/2} (\boldsymbol{x}_n - \boldsymbol{x}_i) \|_2^2) , \tag{4}$$

where $\boldsymbol{C}_n$ is the inverse of the covariance matrix estimated from the gradients in $N(\mathbf{x}_n)$ and $\mathbf{x}_n \in N(\mathbf{x}_i)$.

B. Non-Local Kernel Regression

In NLKR, given the neighborhood $N(\boldsymbol{x}_i)$, the image is searched for similar structures for including in the WLS fit. In other words, self-similarity is explored to further reduce the additive noise. In particular, a non-local term of these similar but non-local neighborhoods $P(\boldsymbol{x}_i)$ is added to (1) [8], which gives rise to the following minimization problem:

$$\begin{aligned} &\min_{\boldsymbol{b}_i} \| \boldsymbol{W}_{K_i} (\boldsymbol{y}_i - \boldsymbol{A}_i \boldsymbol{b}_i) \|_2^2 + \sum_{j \in P(\boldsymbol{x}_i) \setminus \{i\}} \omega_{ij} \boldsymbol{W}_{Kj} \| \boldsymbol{y}_j - \boldsymbol{A}_i \boldsymbol{b}_i \|_2^2 \\ &\Leftrightarrow \min_{\boldsymbol{b}_i} \sum_{j \in P(\boldsymbol{x}_i)} \omega_{ij} \| \boldsymbol{W}_{Kj} (\boldsymbol{y}_j - \boldsymbol{A}_i \boldsymbol{b}_i) \|_2^2 , \end{aligned} \tag{5}$$

where $P(\boldsymbol{x}_i)$ is the collection of patches/neighborhoods with similar structure found through block matching, $\boldsymbol{y}_j$ is the observation of the *j*th similar patch, $\boldsymbol{W}_{K_j}$ is the corresponded weight matrix and ω_{ij} is the corresponding weight, which is usually determined from the similarity between current patch $N(\boldsymbol{x}_i)$ and the *j*th similar patch. Similar to the local case in (3), the estimated pixel value is given by

$$\hat{z}(\boldsymbol{x}_i) = \boldsymbol{e}_1^{\mathrm{T}} [\boldsymbol{A}_i^T (\sum_{j \in P(\boldsymbol{x}_i)} \omega_{ij} \boldsymbol{W}_{K_j}) \boldsymbol{A}_i]^{-1} \boldsymbol{A}_i^T \sum_{j \in P(\boldsymbol{x}_i)} \omega_{ij} \boldsymbol{W}_{K_j} \boldsymbol{y}_j . \tag{6}$$

III. NON-LOCAL KERNEL REGRESSION WITH BANDWIDTH SELECTION

Although the locally adaptive kernel helps to capture of the local structures of the image, one still needs to select the appropriate bandwidth *h*, which is a key parameter in LPR. Usually such parameter is selected empirically [1, 2]. This problem has also been studied extensively in the statistics community and a number of bandwidth selection algorithms are available [3, 6, 10, 11]. Motivated by the usefulness of NLKR, we now study the incorporation of automatic bandwidth selection in the NLKR framework. In particular, we shall study ICI bandwidth selection method due to its effectiveness and relatively low complexity.

The ICI rule was first proposed as an empirical bandwidth selection method for signal filtering in [11]. It has been applied to spatially adaptive nonparametric regression [8], local polynomial regression and modeling [3-7]. The algorithm starts with a set of prescribed possible bandwidth parameters in the ascending order,

$$\boldsymbol{H} = \{h_1 < h_2 < \cdots < h_L\} , \tag{7}$$

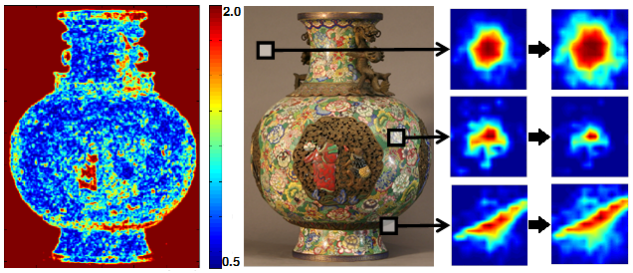
where L is the number of bandwidth candidates. For a certain bandwidth h_i , the confidence intervals $D_k = [L_k, U_k]$ are obtained from the estimated $\hat{\boldsymbol{b}}_i(h_k)$ as

$$U_k = \hat{\boldsymbol{b}}_i(h_k) + \Gamma \sqrt{Var(\hat{\boldsymbol{b}}_i(h_k))} , \tag{8}$$

$$L_k = \hat{\boldsymbol{b}}_i(h_k) - \Gamma \sqrt{Var(\hat{\boldsymbol{b}}_i(h_k))} , \tag{9}$$

where $Var(\cdot)$ denotes the variance, and Γ is a threshold parameter for adjusting the width of the confidence interval. For instance, $\Gamma = 2.58$ implies a 99% confidence interval. A suboptimal bandwidth h_{opt} is determined by checking the intersection of the confidence intervals with the bandwidths $\boldsymbol{H}$ in the ascending order. Therefore, the bandwidth of the last intersected interval is selected as the suboptimal one.

The bandwidth *h* controls the scale of the kernel as shown in (4). We aim to select optimal bandwidths at different locations according to the local structures. Hence not only the shape but also the scale of the kernel will be spatially adaptive across the whole image. The bandwidth of each pixel will be individually selected from the bandwidth set $\boldsymbol{H}$ using the ICI rule. $\hat{\boldsymbol{b}}_i(h_k)$ in (8) and (9) can be estimated by (5) with corresponding weight matrix $\boldsymbol{W}_{K_j}|_{h_k}$. However, $Var(\hat{\boldsymbol{b}}_i(h_k))$ of NLKR has a different form from the one of local kernel



(a) (b) (c) (d)

Figure 1. The kernels with spatially adaptive bandwidth. (a) is an example of the spatially adaptive bandwidth. (b) is the original color image, (c) is the adaptive kernels at the marked blocks without bandwidth selection, (d) is the corresponded kernels with bandwidth selected in (a).

regression, which is found to be

$$Var(\hat{\boldsymbol{b}}_i(h_k)) = Var([\boldsymbol{A}_i^T \boldsymbol{S}_{W_j} \boldsymbol{A}_i]^{-1} \boldsymbol{A}_i^T \boldsymbol{S}_{y_j})|_{h_k}$$
$$= [\boldsymbol{A}_i^T \boldsymbol{S}_{W_j} \boldsymbol{A}_i]^{-1} \boldsymbol{A}_i^T Var(\boldsymbol{S}_{y_j}) \boldsymbol{A}_i [\boldsymbol{A}_i^T \boldsymbol{S}_{W_j} \boldsymbol{A}_i]^{-1}|_{h_k}, \quad (10)$$

$$\boldsymbol{S}_{W_j} = \sum_{j \in P(\boldsymbol{x}_i)} \omega_{ij} \boldsymbol{W}_{K_j}|_{h_k}, \ \boldsymbol{S}_{y_j} = \sum_{j \in P(\boldsymbol{x}_i)} \omega_{ij} \boldsymbol{W}_{K_j} \boldsymbol{y}_j|_{h_k}. \quad (11)$$

For simplicity, we assume that the observations $\boldsymbol{y}_j, j \in P(\boldsymbol{x}_i)$ are independent as they are coming from distant patches, i.e. the deterministic structure is similar but the additive noises are independent. Hence $\boldsymbol{S}_{y_j}$, i.e. the variance of the weighted summation of $\boldsymbol{y}_j$, can be simplified to

$$Var(\boldsymbol{S}_{y_j}) = \sum_{j \in P(\boldsymbol{x}_i)} \omega_{ij}^2 \boldsymbol{W}_{K_j} Var(\boldsymbol{y}_j) \boldsymbol{W}_{K_j}^{\mathrm{T}}|_{h_k}. \quad (12)$$

Since $\boldsymbol{W}_{K_j}|_{h_k}$ is a diagonal matrix, (12) can then be rewritten as the summation of several diagonal matrices,

$$Var(\boldsymbol{S}_{y_j}) = \sum_{j \in P(\boldsymbol{x}_i)} \omega_{ij}^2 \sigma^2(\boldsymbol{y}_j) \boldsymbol{W}_{K_j}^2, \quad (13)$$

where $\boldsymbol{W}_{K_j}^2 = diag[K_{h_k}^2(\boldsymbol{x}_{1,j} - \boldsymbol{x}_j), ..., K_{h_k}^2(\boldsymbol{x}_{p,j} - \boldsymbol{x}_j)]$, $\boldsymbol{x}_{n,j}$ denotes the pixel coordinates in the neighborhood of $\boldsymbol{x}_j$, i.e. $\boldsymbol{x}_{n,j} \in N(\boldsymbol{x}_j)$, and $\sigma^2(\boldsymbol{y}_j)$ is the noise variance of the observation $\boldsymbol{y}_j$.

Using (10)-(13), the confidence intervals D for the bandwidth set $\boldsymbol{H}$ can be determined from (8) and (9). Following the standard procedure of ICI, the suboptimal bandwidth can be determined for each pixel. Fig. 1(a) shows an example of the spatially varying bandwidth across the whole images. The pixel estimate of the proposed bandwidth adaptive NLKR (BA-NLKR) is therefore derived from (6) as

$$\hat{z}(\boldsymbol{x}_i) = \boldsymbol{e}_1^{\mathrm{T}} [\boldsymbol{A}_i^T \boldsymbol{S}_{W_j} \boldsymbol{A}_i]^{-1} \boldsymbol{A}_i^T \boldsymbol{S}_{y_j}|_{h_k}. \quad (14)$$

Fig. 1(b)-(d) illustrates the advantage of the proposed BA-NLKR method by examining typical bandwidth selected in Fig. 1(a). It can be seen that small bandwidths are selected at the region of textures and edges, while larger ones are used at flat area. Fig. 1(c) and 1(d) shows that local structures are better represented with the help of bandwidth selection.

TABEL I. EXECUTION TIME (SECONDS) COMPARISON

Platform	*Block Matching*	*Kernel Construction*	*Regression with Bandwidth Selection*	*Total*
CPU	796.27	16.19	5287.68	6100.14
GPU	**1.75**	**2.07**	**3.34**	**7.16**

IV. GPU IMPLEMENTATION

Though LPR is a flexible framework for denoising and super-resolution, its computational complexity is still high as each pixel has to be estimated individually. The additional non-local similarity searching and bandwidth selection make the algorithm more complicated. Hence, it is highly desirable to accelerate the process by the means of parallel computing.

To this end, we have implemented the proposed algorithm on the GPUs using OpenCL. Three key steps, including block matching, adaptive kernel construction and regression with bandwidth selection, are implemented on GPUs respectively. Since the first two steps are performed at each pixel independently, they can be efficiently parallelized on GPUs. However, the final step needs some modifications because the computation at each pixel is too complicated to be run entirely in the computing units of the GPUs. Hence, this step is divided into three subtasks. Equations (13) and (14) are first parallelly executed on GPU with different bandwidths. Then the ICI rule is applied on the CPU to select the suitable bandwidths. Finally (14) is estimated at each pixel on GPU again using the selected bandwidths. Note that images with high resolution may need to be divided into strips to avoid hitting the time limit of GPU programs if it is executed on a GPU with a display attached.

Table I shows the time required for processing a single image with the resolution of 512×512, while ten nearest blocks and five bandwidth candidates are considered respectively. The CPU implementation is running under the Matlab environment, while the GPU implementation is realized using C and OpenCL. The experiments were carried in a PC with an Intel i920 CPU and an AMD Radeon HD 6950 GPU. It can be seen that the processing time is dramatically reduced to a reasonable level.

V. APPLIACTIONS TO IMAGE AND VIDEO RESTORATION

The BA-NLKR algorithm proposed above provides a general framework for various image and video restoration applications. In this section, we specifically apply (14) to image/video SR and denoising tasks.

A. Image and Video Super-Resolution

Image SR aims to estimate a high-resolution image (HR) from single or several low-resolution images (LR). The LR frames are usually modeled as images blurred and down-sampled from a HR image, i.e.

$$Y_l = D_l H I + N_l = DZ + N_l, \ \ l = 1,2,\ldots., \quad (15)$$

where I denotes a HR image, Y_l denotes the *l-th* LR observations, D_l and H are the corresponding down-sampling and blurring operators respectively, N_i is the noise

Figure 2. Single Frame Super-resolution results (x3, PSNR(dB) in brackets). From left to right: BI (30.9566), NLKR (32.0841), and BA-NLKR (**32.0896**).

term, Z is the blurred HR image which is the target of the estimation. Combining (5) and (15), we can get the formulation for SR,

$$\hat{\boldsymbol{b}}_i = \arg\min_{\boldsymbol{b}_i} \sum_{j\in P(\boldsymbol{x}_i)} \omega_{ij} \left\| D_j^{\mathrm{T}} (\boldsymbol{y}_j - D_j \boldsymbol{A}_i \boldsymbol{b}_i) \right\|_{\boldsymbol{W}_{Kj}}^2, \tag{16}$$

where D_j^{T} denotes the up-sampling operator with zero padding. Therefore, (14) can be reformulated as

$$\hat{z}(\boldsymbol{x}_i) = \boldsymbol{e}_1^{\mathrm{T}} [\boldsymbol{A}_i^T (\sum_{j\in P(\boldsymbol{x}_i)} \omega_{ij} D_j^{\mathrm{T}} D_j \boldsymbol{W}_{K_j} D_j^{\mathrm{T}} D_j) \boldsymbol{A}_i]^{-1} \times \boldsymbol{A}_i^T \sum_{j\in P(\boldsymbol{x}_i)} \omega_{ij} D_j^{\mathrm{T}} D_j \boldsymbol{W}_{K_j} D_j^{\mathrm{T}} \boldsymbol{y}_j |_{h_k} . \tag{17}$$

As we can see, the HR image is recovered by multiple LR patches at different locations with shape and bandwidth varying kernels. The estimated image is then deblurred by a Total Variation-based algorithm [12].

We evaluate the proposed algorithm in both image and video SR tasks. Fig. 2 shows an example of single frame SR compared with Bicubic Interpolation (BI) and NLKR. PSNR comparison for video SR is given in Table II, where the input LR images are zoomed by a factor of 3 and Gaussian noise with a standard deviation of 2 is added. More comparison between the original NLKR algorithm and other state-of-the-art SR algorithms can be found in [9]. Note that the bandwidth is chosen empirically in the former while the bandwidth is chosen automatically in the proposed method. It can be seen that the proposed algorithm has a consistently higher PSNR than NLKR, thanks to the automatic bandwidth selection.

B. Image and Video Denoising

The proposed BA-NLKR algorithm is naturally designed for denoising applications. Hence, (14) can be directly applied to estimate the noise free images and videos. Fig. 3 shows an example of the image denoising results. The noisy images are generated by adding white Gaussian noise with a standard deviation of 20 to the clean ones. The bandwidth of NLKR is manually set to 1 as suggested in [9], while the one of the proposed BA-NLKR is selected automatically. More PSNR comparison can be found in Table II.

VI. CONCLUSIONS

A new BA-NLKR algorithm for image and video restoration is presented. It incorporates the ICI bandwidth selection method into the framework of NLKR to facilitate automatic bandwidth selection so as to achieve better performance. Simulation results show that the proposed algorithm has a higher PSNR than the conventional NLKR method. The proposed algorithm is parallelized in GPU, which provides significant speedup over the conventional CPU implementation. The effectiveness of the proposed algorithm is illustrated by experimental results on both image/video super-resolution and denoising.

Figure 3. Denoising results (PSNR (dB) in brackets). From left to right: Noisy image, NLKR (31.5298), and BA-NLKR (**31.9124**).

TABLE. II PSNR (DB) RESULTS

Algorithms	Video SR		Denoising	
	Foreman	*Miss America*	*Peppers*	*Cameraman*
NLKR	33.8274	35.8325	29.9053	28.6629
BA-NLKR	**33.8311**	**35.8372**	**30.6681**	**29.5160**

REFERENCES

[1] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel Regression for Image Processing and Reconstruction," *IEEE Trans. Image Process.*, Vol. 16, No. 2, 349-366, Feb. 2007.

[2] H. Takeda, P. Milanfar, M. Protter, and M. Elad, "Super-Resolution Without Explicit Subpixel Motion Estimation," *IEEE Trans. Image Process.*, Vol. 18, No. 9, 1958-1975, Sep. 2009.

[3] Z. G. Zhang, S. C. Chan, K. L. Ho, and K. C. Ho, "On Bandwidth Selection in Local Polynomial Regression Analysis and Its Application to Multi-Resolution Analysis of Non-Uniform Data," *J. Signal Process. Syst.*, vol. 52, no. 3, pp. 263-280, Sept. 2008.

[4] Z. G. Zhang and S. C. Chan, "On Kernel Selection of Multivariate Local Polynomial Modelling and Its Application to Image Smoothing and Reconstruction," *J. Signal Process. Syst.*, Vol. 64, No. 3, pp. 361-374, 2011.

[5] Z. Zhang, Y. S. Hung, and S. C. Chan, "Local Polynomial Modeling of Time-Varying Autoregressive Models with Application to Time-Frequency Analysis of Event-related EEG," *IEEE Trans. Biomed. Eng.*, Vol. 58, No. 3, pp. 557-566, Mar. 2011.

[6] S. C. Chan and Z. Zhang, "Local Polynomial Modeling and Variable Bandwidth Selection for Time-Varying Linear Systems," *IEEE Trans. Instrum. Meas.*, Vol. 60, No. 3, pp. 1102-1117, Mar. 2011.

[7] Z. G. Zhang, S. C. Chan, and C. Wang, "A New Regularized Adaptive Windowed Lomb Periodogram for Time-Frequency Analysis of Nonstationary Signals With Impulsive Components," *IEEE Trans. Instrum. Meas.*, Vol. 61, No. 8, 2283-2304, 2012.

[8] V. Katkovnik, K. Egiazarian, and J. Astola, "A spatially adaptive nonparametric regression image deblurring," *IEEE Trans. Image Process.*, Vol. 14, No.10, 1469–1478, 2005.

[9] H. Zhang, J. Yang, Y. Zhang and T. S. Huang, "Non-Local Kernel Regression for Image and Video Restoration," The 11th European Conference on Computer Vision (ECCV), 2010.

[10] J. Fan and I. Gijbels, "Data-Driven Bandwidth Selection in Local Polynomial Fitting: Variable Bandwidth and Spatial Adaptation," Statistica Sinica, Vol. 57, pp. 371-394, 1995.

[11] V. Katkovnik, "A new method for varying adaptive bandwidth selection," *IEEE Trans. Signal Process.*, Vol. 47, No. 9, pp. 2567-2571, 1999.

[12] Pascal Getreuer, http://www.getreuer.info/home/tvreg, 2009.